Improving TCP Performance over 802.11 WLAN with Radio Network Feedback

A. Jayananthan
Electrical and Computer Engineering
University of Canterbury
New Zealand
a.nanthan@elec.canterbury.ac.nz

Harsha R. Sirisena
Electrical and Computer Engineering
University of Canterbury
New Zealand
h.sirisena @elec.canterbury.ac.nz

Abstract**—** A new technique is proposed that enables TCP to distinguish wireless packet losses from congestion related packet losses. This is achieved by introducing a wireless enhancement proxy (WENP) at layer 3 in the base station that monitors the TCP packet flows in both directions, detects wireless packet drops and provides feedback to the TCP sender using one of the TCP header reserved bits. The standard TCP Reno congestion control mechanism is modified to accommodate this effect. The scheme is modeled using the OPNET tool, and the simulation results are presented and explained. The performance of the proposed scheme is compared with that of a standard WLAN without a proxy. Our results show that introducing WENP significantly improves the TCP performance in a WLAN environment.

Keywords- TCP Throughput, WLAN, Wireless Packet Loss, Wireless Enhancement Proxy

I. INTRODUCTION

One of the most challenging and interesting trends in recent computer networks is the integration of mobile communications. Wireless local area networks (WLANs) based on the IEEE 802.11 MAC protocol [1] are becoming ubiquitous as they can deliver services commonly found in wired networks. However, the performance of transport layer protocols, such as TCP[2], over 802.11 may be degraded considerably due to the characteristics of the wireless medium that suffers from significantly high bit error rates.

Transmission Control Protocol (TCP) is primarily designed for wired networks where the intrinsic bit error rate is negligible and the main cause of packet loss is network congestion. TCP suffers a significant degradation in performance over wireless networks because the assumptions behind TCP do not hold in a wireless environment. However, being dominant in wired networks and due to the desire for internetworking between wired and wireless networks, TCP has become the most commonly used transport protocol in mobile Internet technology. TCP should therefore be redesigned with the problems of the wireless environment in mind.

A number of approaches, such as Performance Enhancing Proxies[3], Explicit Congestion Notification (ECN)[4], link-layer protocols[5], split connection protocols [2, 6] and the Snoop protocol [7] have been proposed to improve the TCP performance over an unreliable wireless network.

In this paper, we present a technique that detects wireless packet losses and enables the TCP sender to completely distinguish wireless packet losses from congestion related losses. It also helps the TCP sender to minimize the unnecessary TCP timeouts leading to a reduction in the congestion window when non-congestion related losses occur.

The rest of the paper is organized as follows. An overview of TCP and WLAN is given in Sections II and III, respectively. Previous solutions to improve TCP performance over wireless links are surveyed in Section IV. The proposed scheme and its OPNET implementation are briefed in Section V. The network model and simulation results are presented in Section VI and VII, respectively. Finally, the conclusions drawn and ideas for future work are presented in Section VIII.

II. TCP OVERVIEW

TCP is a reliable, connection-oriented protocol that adapts to the network by dynamically learning the delay characteristics of a network and adjusting its operation to maximize throughput without overloading the network. It has four phases: slow start, congestion avoidance, fast recovery and fast retransmit. On opening a connection, the TCP sender enters the slow start phase in which the congestion window (cwnd) is increased by one maximum size segment (MSS) per acknowledgement (ack) received. Exponential growth ensues until cwnd reaches the slow-start threshold (ssthresh) when it enters the congestion avoidance phase.

During congestion avoidance, cwnd is incremented by 1 MSS per round trip time (RTT) and this phase continues until congestion is detected. When a TCP sender detects segment loss using the retransmission timer, it will enter into slow start. Else fast retransmit is triggered if the sender receives a third duplicate acknowledgment (dupack) before timeout occurs. It then retransmits the lost segment, saves the slow-start threshold as half of the current cwnd and increments cwnd by 1 MSS for each additional dupack. When the next ack arrives that acknowledges new data, cwnd is set to the slow-start threshold and the TCP fast recovery phase ends. Notice that the sender reduces its cwnd, or enters into slow start, even if the segment loss is due to a temporary wireless effect rather than congestion.

III. WLAN OVERVIEW

An 802.11 LAN is based on a cellular architecture where the system is subdivided into cells. Each cell, called Basic Service Set (BSS), is controlled by a Base Station called Access Point (AP). When a WLAN is formed from several BSSs, APs are connected through a backbone network, typically Ethernet, called a Distribution System (DS). The whole interconnected WLAN including the different BSSs with their respective APs and the DS is called an Extended Service Set (ESS). The 802.11 protocol covers the MAC and Physical layers; the Standard currently defines a single MAC that interacts with the Physical layer using Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum or Infrared.

Besides the standard MAC layer functionality, the 802.11 MAC performs other functions, such as fragmentation, packet retransmissions and acknowledges. The MAC defines two different access methods: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The DCF is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. It works as follows; a station willing to transmit senses the medium, if the medium is busy it defers its transmission. If the medium is free for a specified time called Distributed Inter Frame Space (DIFS) then the station is allowed to transmit. On receiving the packet successfully, the receiver will send an ack. The receipt of the ack will indicate to the transmitter that no collision has occurred. If the receiver does not receive the ack, it will keep retransmitting the packet until it either gets acknowledged or is discarded after a certain number of retransmissions.

IV. RELATED WORK

This Section briefly explains some of the protocols previously proposed to improve TCP performance over wireless links.

The AIRMAIL protocol [5] makes the link layer reliable in conjunction with forward error correction (FEC). The base station sends an entire window of data before an ack is returned in order to conserve bandwidth (BW) and power. This approach does not allow error correction until the end of the window, causing the TCP sender to timeout if the error rate is high, resulting in poor TCP throughput.

The split connection scheme [2, 6] handles the problems due to mobility and unreliability of a wireless link entirely within the wireless link and shields the TCP sender from them. This approach adds the overhead of going through TCP protocol processing twice at the base station. Moreover, timeouts on the wireless link cause the original TCP sender to timeout, resulting in poor end-to-end throughput. It also fails on the basic principle of not violating the TCP end-to-end semantic.

The Snoop protocol [7] introduces a module at the base station that uses link level buffers to cache packets passing across the wireless link. It retransmits unacknowledged packets locally and filters out the dupacks to hide the wireless loss from the TCP sender. This helps TCP to avoid unnecessary fast retransmissions, or worse still, triggering the congestion control mechanism. However, it cannot completely shield the TCP sender from wireless losses[8].

ECN [4] provides feedback to TCP sources about congestion at routers in the network before packets are dropped at congested routers. It adds two new flags in the reserved field of the TCP header designated as the ECN Echo (ECE) flag, set by the TCP receiver to indicate congestion in the network, and the Congestion Window Reduced (CWR) flag, set by the TCP sender, to inform the TCP receiver that it has reacted to its congestion notification. ECN requires support from both routers and end hosts.

The aforementioned protocols use two different approaches to improve the TCP performance over wireless links; one is to hide non-congestion related wireless losses from the TCP sender and the other is to make the TCP sender aware of these losses. However, they all fail to completely distinguish the wireless losses from the congestion related losses. We propose a new scheme that makes the TCP sender completely aware of wireless losses and to react accordingly.

V. PROPOSED SCHEME AND OPNET IMPLEMENTATION

In previous studies, TCP performance has been improved essentially by enhancing link layer protocols. We adopt the alternative view that TCP improvements should be achieved by tuning TCP itself to utilize the available network resources efficiently in both wireline and wireless environments. Our proposed scheme introduces a proxy, called WENP that monitors the WLAN network radio interface and, with the aid of one of the TCP header reserved bits, notifies the TCP sender of any effects caused by the wireless link. The TCP end-to-end semantic is maintained but it is modified in order to adapt to the characteristics of the wireless environment.

A. Wireless Enhancement Proxy (WENP)

WENP is introduced between the WLAN MAC and the IP layers of WLAN Ethernet router. On receiving data packets from the IP layer, it obtains the TCP header information, from the IP datagram, assuming it is not encrypted, and maintains them in a cache table. It also keeps a connection table in order to be able to support multiple TCP connections. Figure 1 shows the simplest flow control to extract the TCP header information from IP data packets.

On the arrival of ack packets from the MAC layer, WENP uses both the cache table and connection table to detect wireless packet losses. If a wireless packet loss is detected, it notifies the TCP sender of this effect by utilizing the control bit next to the CWR flag in the reserved field of the TCP header, called Radio Network Feedback (RNF) flag. Figure 2 shows the simplest flow control for wireless packet loss detection.

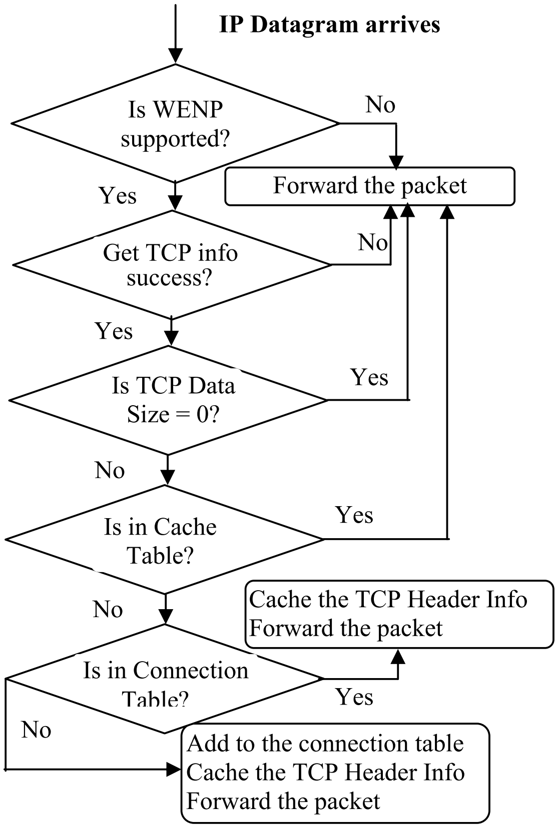

Figure 1. Simplest flow control to extract TCP header

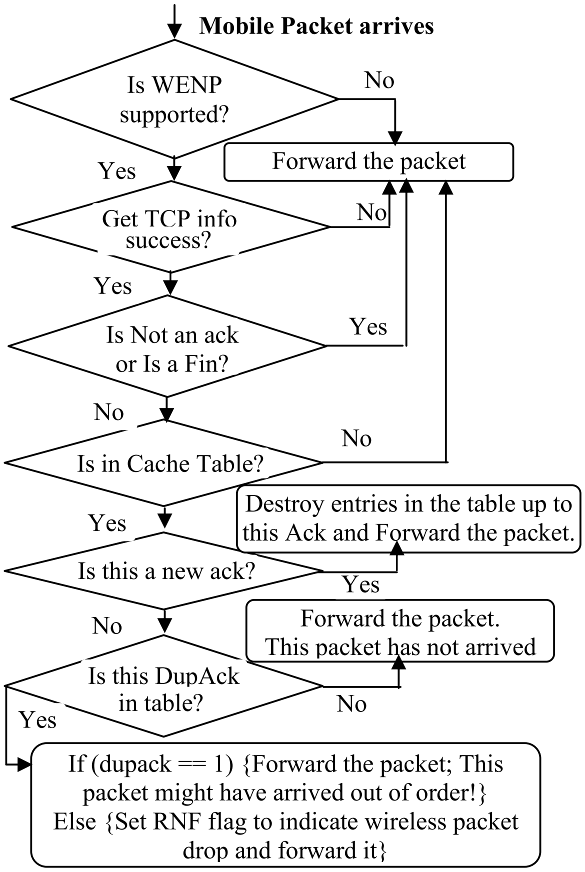

Figure 2 Simplest flow control for wireless packet loss detection

B. TCP Reno Modifications

In standard TCP Reno, the TCP sender concludes that the network has dropped a packet when it receives three dupacks. It then immediately retransmits the dropped packet without waiting for the retransmission timer to expire. The rationale is that the sooner the fast retransmit occurs, the better TCP performs because it avoids unnecessary TCP timeouts. With this in mind, and the fact that TCP sender can confirm any wireless packet drops when it receives dupacks with the RNF flag set, the TCP Reno is fine tuned to retransmit the packets dropped across the wireless link when it receives two dupacks with the RNF flag set. It should be noticed that the TCP sender, considering the delayed packet arrivals at the base station, still waits for two dupacks with RNF flag set even though it has received confirmation that the packet has been dropped across the wireless link.

Now it comes to deciding whether to reduce the cwnd when TCP sender detects wireless packet drops. We define α to be the BW utilization factor during the fast retransmit and recovery phase. The value of α can be between 0.5 and 1. In TCP Reno and New Reno, α is assigned the value 0.5; they halve the cwnd when a packet drop is detected by three dupacks arrival. The value for α can be optimized for wireless packet recovery, but it is out of scope of this paper and we judiciously assign α the value 0.75. This will increase the TCP throughput by cwnd/4 per RTT after each wireless packet recovery, if the receiver window is bigger than the cwnd. It will also increase the chance of recovering multiple packet losses within a window of data, which are highly likely in wireless environments; to recover two packet drops, it requires cwnd $> (N+3)/\alpha$, where N is the number of packet drops.

VI. OPNET NETWORK MODEL

The network model used for this study is shown in Figure 3. The LAN is extended using a WLAN Ethernet router that forms a WLAN together with some mobile hosts (MHs). The model consists of two Servers, and some fixed hosts (FHs) and MHs. Servers and FH clients are connected to the WLAN router through switches using 10baseT point-to-point link model. The Application and Profile Configuration nodes are configured to generate different applications such as HTTP, FTP Database and Email. FH clients are configured to utilize some of these services in parallel with MH clients in order to make the network analyzes be realistic.

All MH clients are configured to download FTP files from Server-1 simultaneously. Packets coming from the Server-1 are dropped at the MAC layer of MH clients, using a uniform probability distribution. Table I shows the packet drop rates of MHs. The WLAN Router is implemented, in turn, with a standard WLAN and a WLAN with WENP to compare their relative performances. Modified TCP Reno with the default parameters [9] is used in all simulation scenarios. Selected TCP Reno and WLAN parameter values are given in Tables II and III, respectively.

TABLE I. UES CONFIGURATIONS

MHs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Packet drops (%)	0	2	4	6	8	10	15	20

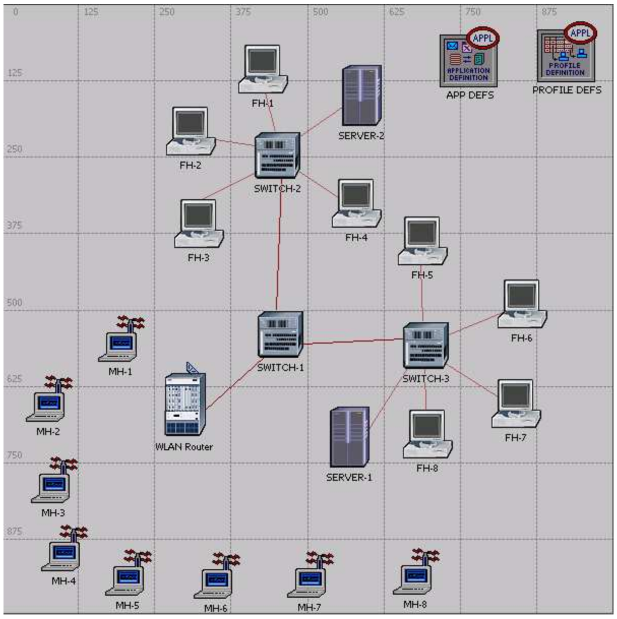

Figure 3. Network Model

TABLE II. TCP RENO PARAMETERS

Receive Window Size at FTP Server-1	65535 bytes
Receive Window Size at UEs	Default
Delay ACK Mechanism	Segment/Clock Based
Maximum ACK Delay	0.2 seconds
Maximum ACK Segments	2
Maximum Segment Size (MSS)	1460 bytes
Sack Option	Disabled
Slow-Start Initial Count	1 MSS
Minimum RTO	1 seconds
Maximum RTO	64 seconds

TABLE III. WLAN PARAMETERS

Physical Characteristics	Direct Sequence
Data Rate	11 Mbps
Packet Reception-Power Threshold	-95 dBm
Short Retry Limit	7
Long Retry Limit	4
PCF Parameters	Disabled

VII. SIMULATION RESULTS AND DISCUSSION

Figure 4 shows the TCP cwnd size, sent segment sequence number, the MAC packet drops and the number of cached TCP headers responses during FTP file upload by MH-4 client with our proposed scheme, WLAN with WENP, and the standard WLAN. Figure 5 shows a snapshot of TCP cwnd response for MH-4. Comparison of the TCP sent segment sequence number responses for MH-5, MH-6, MH-7 and MH-8 is given in Figure 6.

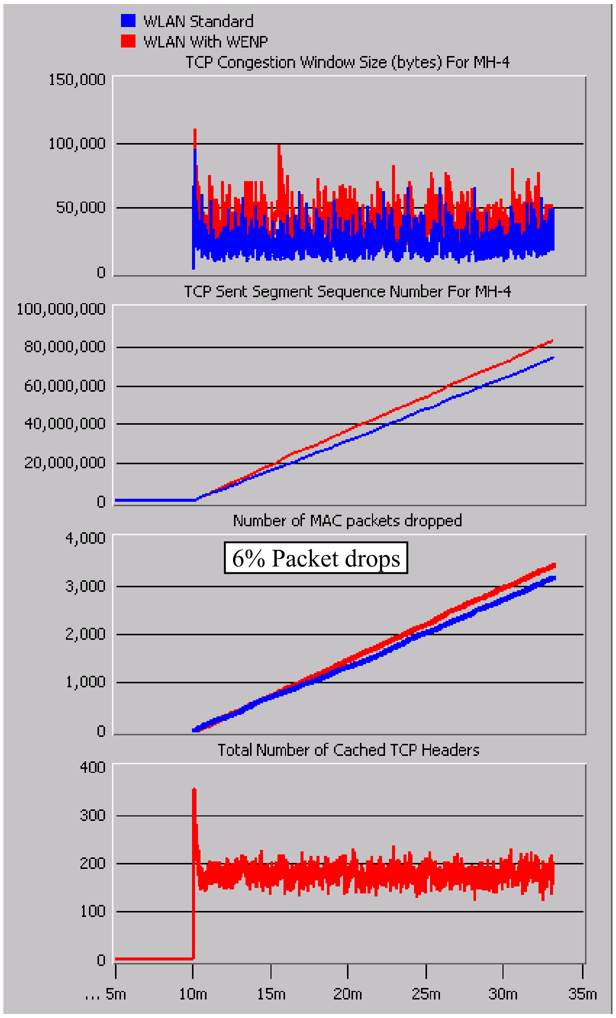

Figure 4. Responses during MH-4 client FTP file upload

From Figures 4-6, it can be seen that WENP does not add significant overhead to the WLAN Router and significantly improve the cwnd and throughput responses compared to that of the standard TCP. The simulation was repeated many times with different seed values, which generates different MAC packet drop patterns, and the TCP mean throughput value was obtained with less than 5% error margin. Table IV summarizes the TCP throughput performances. Figure 7 shows the TCP throughput improvement with our proposed scheme versus the MAC packet drop rates.

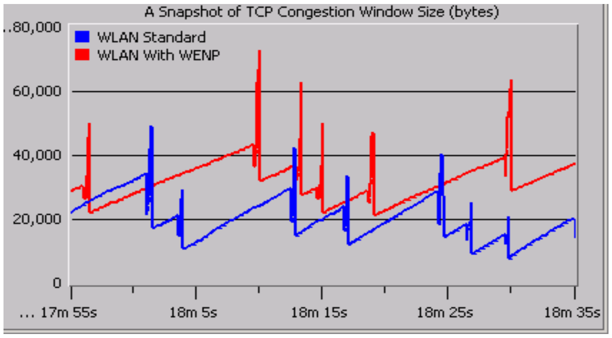

Figure 5. A Snapshot of TCP cwnd response

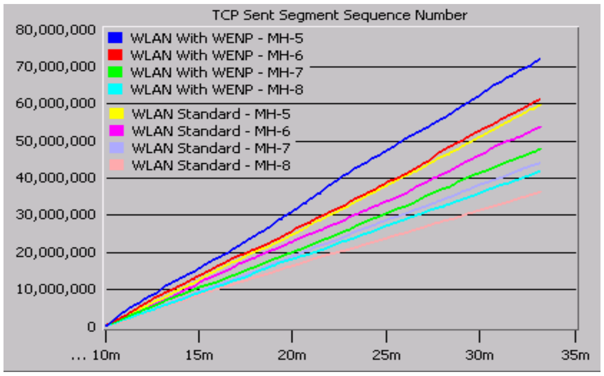

Figure 6. Comparison of TCP sent segment responses

TABLE IV. SUMMARY OF TCP THROUGHPUT PERFORMANCE

TCP throughput (Kbits/sec)	MH							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Standard WLAN	1480	747	524	429	356	307	251	214
WLAN with WENP	1246	806	569	472	406	349	278	234

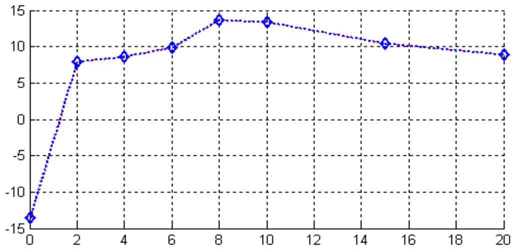

Figure 7. TCP Throughput improvement Vs Packet drop rates

From Table 4 and Figure 7, it can be observed that our proposed scheme has improved the TCP throughput performance of MHs with MAC packet drops. It should also be noticed that our proposed scheme has decreased the TCP throughput performance of MH-1, which does not drop any MAC packets. It demonstrates that modified TCP fast retransmit and recovery algorithms to handle wireless packet drops perform much fairer than does the standard TCP Reno implementation. With our proposed scheme, the available BW across the wireless medium is fairly shared among the MHs.

VIII. CONCLUSIONS AND FUTURE WORK

A new scheme, WENP, was presented that detects and distinguishes wireless packet losses from congestion- related packet losses. WENP was implemented in a WLAN model in OPNET. The TCP Reno was modified to handle wireless packet losses with the aid of the RNF flag. Simulation results showed that WENP improved the TCP performance significantly compared to that of the standard WLAN. It also enabled the modified TCP Reno to distinguish wireless packet losses from the congestion-related losses and to trigger the wireless enhanced fast retransmit and fast recovery mechanisms to quickly recover from wireless packet losses.

Simulation results also showed that WENP can handle multiple nodes and multiple TCP connections, and utilized the available network resources efficiently and fairly by adapting to the network characteristics. WENP does not inject any additional packet into the network to provide feedback when it detects wireless packet losses; it only sets the RNF flag. We are currently implementing the WENP scheme with other TCP flavors to investigate whether they, too, would benefit from it.

REFERENCES

[1] B. O'Hara and A. Petrick, *IEEE 802.11 Handbook : -- a designer's companion*: New York : Standards Information Network, IEEE, c2005.

[2] "Transmission Control Protocol," RFC 761, January 1980. http://www.faqs.org/rfcs/rfc761.html.

[3] "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135, June 2001. http://www.faqs.org/rfc/rfc3135.txt.

[4] R. Ramani and A. Karandikar, "Explicit congestion notification (ECN) in TCP over wireless network," 2000.

[5] S. P. Ender Ayanoglu, Thomas F. LaPorta, Krishan K. Sabnani, Richard D. Gitlin, "AIRMAIL: A Link-Layer Protocol for Wireless Networks," ACM Wireless Networks, February 1995.

[6] A. Bakre and B. R. Badrinath, "I-TCP: indirect TCP for mobile hosts," 1995.

[7] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *ACM Wireless Networks Journal 1, 4, 469–481.*

[8] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," ACM Press New York, NY, USA

[9] "OPNET Technologies, Inc.," http://www.opnet.com/.